AN EFFICIENT MESHFREE IMPLICIT FILTER FOR NONLINEAR FILTERING PROBLEMS[*]

FENG BAO[†], YANZHAO CAO[‡], CLAYTON G. WEBSTER[†], AND GUANNAN ZHANG[†]

Abstract. In this paper, we propose a meshfree approximation method for the implicit filter developed in [2], which is a novel numerical algorithm for nonlinear filtering problems. The implicit filter approximates conditional distributions in the optimal filter over a deterministic state space grid and is developed from samples of the current state obtained by solving the state equation implicitly. The purpose of the meshfree approximation is to improve the efficiency of the implicit filter in moderately high-dimensional problems. The construction of the algorithm includes generation of random state space points and a meshfree interpolation method. Numerical experiments show the effectiveness and efficiency of our algorithm.

Key words. Nonlinear filtering, implicit algorithm, meshfree approximation, Shepard's method

1. Introduction.

Nonlinear filters are important tools for dynamical data assimilation with applications in a variety of research areas, including biology [1, 20], mathematical finance [4, 11], signal processing [14, 20, 23], image processing [22], and multi-target tracking [15, 18]. To put it succinctly, nonlinear filtering is an extension of the Bayesian framework to the estimation and prediction of nonlinear stochastic dynamics. In this effort, we consider the following nonlinear filtering model

$$\begin{cases} \dfrac{dX_t}{dt} = f(t, X_t; W_t), & \text{(state)} \\ \quad Y_t = g(t, X_t) + V_t, & \text{(observation)} \end{cases} \tag{1.1}$$

where f and g are two nonlinear functions, $\{X_t \in \mathbb{R}^d, t \geq 0\}$ and $\{Y_t \in \mathbb{R}^q, t \geq 0\}$ are the stochastic state and observation processes, respectively, $\{W_t \in \mathbb{R}^r, t \geq 0\}$ is a random vector representing the uncertainty in X_t, and $\{V_t \in \mathbb{R}^s, t \geq 0\}$ denotes the random measurement error in Y_t. In the discrete setting, the nonlinear filtering model in (1.1) takes the form

$$\begin{cases} X_k = f_k(X_{k-1}, w_{k-1}), & \text{(state)} \\ Y_k = g_k(X_k) + v_k, & \text{(observation)} \end{cases} \tag{1.2}$$

where $\{w_k\}_{k \in \mathbb{N}^+} \in \mathbb{R}^r$ and $\{v_k\}_{k \in \mathbb{N}^+} \in \mathbb{R}^s$ are mutually independent white noises. Let $Y_{1:k} := \sigma\{Y_1, Y_2, \cdots, Y_k\}$ denote the σ filed generated by the observational data up to the step k. The goal of nonlinear filtering is to find the posterior probability density function (PDF) of the state X_k, given the observation data $Y_{1:k}$, so as to compute the quantity of interest (QoI), given by

$$\mathbb{E}[\Phi(X_k)|Y_{1:k}] = \inf\left\{\mathbb{E}[|\Phi(X_k) - Z|^2]; Z \in \mathcal{Z}_k\right\},$$

[*] This material is based upon work supported in part by the U.S. Air Force of Scientific Research under grant numbers 1854-V521-12 and FA9550-11-1-0149; by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract and award numbers ERKJ259, ERKJE45, and de-sc0010678; and by the Laboratory Directed Research and Development program at the Oak Ridge National Laboratory, which is operated by UT-Battelle, LLC., for the U.S. Department of Energy under Contract DE-AC05-00OR22725.

[†] Department of Computational and Applied Mathematics, Oak Ridge National Laboratory, Oak Ridge, TN 37831 (baof@ornl.gov, webstercg@ornl.gov, zhangg@ornl.gov).

[‡] Department of Mathematics and Statistics, Auburn University, Auburn, Alabama, 36849 (yzc0009@auburn.edu).

where $\Phi(\cdot)$ is a test function, and $\mathcal{Z}_k$ denotes the space of all $Y_{1:k}$-measurable and square integrable random variables.

Tremendous efforts have been made to solve nonlinear filtering problems in the last few decades. Two of the well-known Bayesian filters are extended Kalman filters (EKFs) [3,10,16,17,19], and particle filters [5,7,8,13]. The key ingredient of the EKFs is the linearization of both f and g in (1.1), so that the standard Kalman filter can be applied directly. Thus, if the nonlinearity of the state and the observation systems is not severe, then the EKFs can provide efficient and reasonable inferences about the state, otherwise, the performance of the EKFs can be very poor. For particle filters, the central theme is to approximate the desired posterior PDF of the state by the empirical distribution of a set of adaptively selected random samples (referred to as "particles"). The particle filter method is essentially a sequential Monte Carlo approach, which requires no assumption on the linearity of the underlying system. As such, with sufficiently large number of particles, it is capable of providing an accurate approximation of the posterior PDF for a highly nonlinear filtering problems. However, there are some fundamental issues concerning the efficiency and robustness of particle filters [9]. For example, since the empirical PDF is constructed based on particles with equal weights after resampling, the particle filter still needs a lot of samples in order to accurately approximate the target distribution.

To overcome such a disadvantage, the authors proposed a new nonlinear filter named "implicit filter" [2] . This approach adopts the framework of Bayesian filtering, which has two stages at each time step, i.e., prediction and update. At the prediction stage, we estimate the prior PDF $p(X_k|Y_{1:k-1})$ of the future state X_k given the current available observation information $Y_{1:k-1}$; at the update stage, we update the prior PDF by assimilating the newly received data Y_k to obtain the estimate of the posterior PDF $p(X_k|Y_{1:k})$. The implicit filter is distinguished from the particle filters by the use of interpolatory approximations to the prior and posterior PDFs. Specifically, in the particle filter, $p(X_k|Y_{1:k-1})$ is approximated by *explicitly* propagating the samples of the current state $X_{k-1}|Y_{1:k-1}$ through the nonlinear state equation $X_k = f_k(X_{k-1}, w_{k-1})$, and constructing the empirical PDF of $X_k|Y_{1:k-1}$. In the implicit filter, the interpolation of $p(X_k|Y_{1:k-1})$ requires its function values at a set of grid points of the future state X_k. Under the condition that $X_k = x \in \mathbb{R}^d$, we solve *implicitly* the state equation $x = f_k(X_{k-1}, w_{k-1})$ given a set of Monte Carlo samples of w_{k-1}, so that the value of $p(X_k = x|Y_{1:k-1})$, at the grid point of x, can be estimated by averaging the function values of $p(X_{k-1}|Y_{1:k-1})$ at all the solutions of the state equation. As an implicit scheme, the implicit filter has a stabilizing effect which provides more accurate numerical approximations to the solution of the nonlinear filtering problem than the particle filter method [2].

The main challenge of the implicit filter method is that the conditional PDF of the nonlinear filtering solution is estimated at grid points. As such the method suffers the so called "the curse of dimensionality" when the dimension of the state variable is high. In addition, the efficiency of the method may be significantly reduced when the domain of the PDF is unbounded. In this paper, we propose to construct a meshfree implicit filter algorithm to alleviate the aforementioned challenges. Motivated by the particle filter method, we first generate a set of random particles and propagate these particles through the system model and use these particles to replace the grid points in the state space. After that we generate other necessary points through the Shepard's method which constructs the interpolant by the weighted average of the values on state points [12]. In order to prevent particle degeneracy in the generation of random state points, we introduce a resample step in the particle propagation. In addition we choose state points according to the system state, which make them adaptively located in the high probability region of the PDF of state. In this way, we solve the nonlinear filtering problem in a relatively small region in the state space at each time step and approximate the solution on a set of meshfree state points distributed adaptively to the desired PDF of the state. Furthermore, since we approximate the PDF as a function on each state point, instead of using state points themselves to describe the empirical distribution, the implicit filter algorithm requires much fewer points than the particle filter method to depict the PDF of the state.

The rest of this paper is organized as follows. In §2, we introduce the mathematical framework of the Bayesian optimal filter. In §3, we construct meshfree implicit algorithm. In §4, we demonstrate the efficiency and accuracy of our algorithm through numerical experiments. Finally, §5 contains conclusions and directions for the future research.

2. Bayesian optimal filter. For $m, n \in \mathbb{N}^+$, let $X_{m:n}$ and $Y_{m:n}$ denote the σ fields generated by $\{X_m, X_{m+1}, \dots, X_n\}$ and $\{Y_m, Y_{m+1}, \dots, Y_n\}$, respectively. For $k = \mathbb{N}^+$, we use x_k to represent a realization of the random variable X_k, and define

$$p(x_k|\cdot) := p(X_k = x_k|\cdot)$$

for notational simplicity. It is easy to see that the dynamical model in (1.2) is Markovian in the sense that

$$p(x_k|X_{1:k-1}, Y_{1:k-1}) = p(x_k|X_{k-1}).$$

We also know that the measurements are conditionally independent given x_k, i.e.,

$$p(Y_k|X_{1:k}, Y_{1:k-1}) = p(Y_k|x_k).$$

The Bayesian optimal filter constructs the conditional distribution $p(x_k|Y_{1:k})$ recursively in two stages: prediction stage and update stage.

For $k = 1, 2, \cdots$, assume that $p(x_{k-1}|Y_{1:k-1})$ is given. In the prediction stage $p(x_k|Y_{1:k-1})$ is evaluated through the Chapman-Kolmogorov formula:

$$p(x_k|Y_{1:k-1}) = \int_{\mathbb{R}^d} p(x_k|x_{k-1}) p(x_{k-1}|Y_{1:k-1}) dx_{k-1}. \tag{2.1}$$

In the update stage, the prior PDF obtained in (2.1) is used to obtain the posterior PDF $p(x_k|Y_{1:k})$ via the Bayes' formula:

$$p(x_k|Y_{1:k}) = \frac{p(Y_k|x_k)p(x_k|Y_{1:k-1})}{p(Y_k|Y_{1:k-1})} = \frac{p(Y_k|x_k)p(x_k|Y_{1:k-1})}{\int_{\mathbb{R}^d} p(Y_k|x_k)p(x_k|Y_{1:k-1})\, dx_k}. \tag{2.2}$$

3. The meshfree implicit filter. In this section, we construct the meshfree implicit filter algorithm. The algorithm is based the implicit filter algorithm on grid points [2]. The implicit filter algorithm introduced in [2] is developed from the general framework of the Bayesian optimal filter discussed above, in which the primary computational challenge is the numerical approximation of the term $p(x_k|x_{k-1})$ in (2.1).

3.1. The prediction stage. For $k = 1, 2, \cdots$, the goal of this stage is to approximate the prior distribution $p(x_k|Y_{1:k-1})$ of the state X_k, given the posterior distribution $p(x_{k-1}|Y_{1:k-1})$ of the state X_{k-1}. Due to the the fact that

$$p(x_k|x_{k-1}) = \mathbb{E}_w[p(x_k|x_{k-1}, w_{k-1})] = \int_{\mathbb{R}^r} p(x_k|x_{k-1}, w_{k-1}) \cdot p(w_{k-1}) dw_{k-1},$$

the prior PDF $p(x_k|Y_{1:k-1})$ derived in identity (2.1) can be rewritten as

$$p(x_k|Y_{1:k-1}) = \int_{\mathbb{R}^d} \mathbb{E}_w[p(x_k|x_{k-1}, w_{k-1})] p(x_{k-1}|Y_{1:k-1}) dx_{k-1}, \tag{3.1}$$

where $\mathbb{E}_w[\cdot]$ represents the expectation with respect to the white noise w_{k-1}, and the PDF $p(x_k|x_{k-1}, w_{k-1})$ is

$$p(x_k|x_{k-1}, w_{k-1}) = \begin{cases} \infty, & x_k = f_k(x_{k-1}, w_{k-1}), \\ 0, & x_k \neq f_k(x_{k-1}, w_{k-1}), \end{cases} \tag{3.2}$$

with $\int_{\mathbb{R}^d} p(x_k|x_{k-1}, w_{k-1}) dx_k = 1$ for any $x_{k-1} \in \mathbb{R}^d$ and $w_{k-1} \in \mathbb{R}^r$. The definition in (3.2) can be viewed as a generalization of the Dirac delta function in the space $\mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^r$, where the mass is located according to the state equation $x_k = f_k(x_{k-1}, w_{k-1})$.

Note that the estimation of (3.1) requires an approximation to the expectation $\mathbb{E}_w[p(x_k|x_{k-1}, w_{k-1})]$. To this end, we first draw M independent samples $\{w_{k-1}^j\}_{j=1}^M$ of the white noise w_{k-1}, and define an approximation to $p(x_k|x_{k-1}, w_{k-1})$ as

$$\pi^M(x_k|x_{k-1}, w_{k-1}) := \sum_{j=1}^{M} \delta_{w_{k-1}^j}(x_k|x_{k-1}, w_{k-1}), \tag{3.3}$$

with

$$\delta_{w_{k-1}^j}(x_k|x_{k-1}, w_{k-1}) := \begin{cases} \infty, & w_{k-1} = w_{k-1}^j \text{ and } x_k = f_k(x_{k-1}, w_{k-1}^j), \\ 0, & \text{otherwise}, \end{cases}$$

which is essentially a restriction of $p(x_k|x_{k-1}, w_{k-1})$ in the subset $\{w_{k-1}^j\}_{j=1}^M$. Therefore, the expectation $\mathbb{E}_w[p(x_k|x_{k-1}, w_{k-1})]$ in (3.1) can be approximated by

$$\begin{aligned} \mathbb{E}_w[p(x_k|x_{k-1}, w_{k-1})] &\approx \mathbb{E}_w\left[\pi^M(x_k|x_{k-1}, w_{k-1})\right], \\ &= \sum_{j=1}^{M} \int_{\mathbb{R}^r} \delta_{w_{k-1}^j}(x_k|x_{k-1}, w_{k-1}) p(w_{k-1}) dw_{k-1}. \end{aligned} \tag{3.4}$$

To construct an interpolation of $p(x_k|Y_{1:k-1})$, the next step is to approximate $p(x_k|Y_{1:k-1})$ at a point set $\mathcal{H}_k := \{x_k^i\}_{i=1}^N \subset \mathbb{R}^d$ with $N \in \mathbb{N}^+$. By substituting $x_k = x_k^i$ into (3.1)-(3.4), we have

$$p(x_k^i|Y_{1:k-1}) = \int_{\mathbb{R}^d} \mathbb{E}_w\left[\pi^M(x_k^i|x_{k-1}, w_{k-1})\right] p(x_{k-1}|Y_{1:k-1}) dx_{k-1} + \mathcal{R}_{k|k-1}^i, \tag{3.5}$$

where $\mathcal{R}_{k|k-1}^i := p(x_k^i|Y_{1:k-1}) - \int_{\mathbb{R}^d} \mathbb{E}_w[\pi^M(x_k^i|x_{k-1}, w_{k-1})] p(x_{k-1}|Y_{1:k-1}) dx_{k-1}$ is the approximation error. Then, by further fixing $w_{k-1} = w_{k-1}^j$, the location of the mass of $\delta_{w_{k-1}^j}(x_k^i|x_{k-1}, w_{k-1}^j)$ in the space of x_{k-1}, denoted by $x_{k-1}^{i,j}$, can be obtained by *implicitly* solving the state equation

$$f_k\left(x_{k-1}^{i,j}, w_{k-1}^j\right) = x_k^i, \quad j = 1, \cdots, M,$$

which is the reason we named the approach the implicit filter. Now substituting x_k^i into (3.4), and using the same sample set $\{w_{k-1}^j\}_{j=1}^M$ as above to approximate the integral on the right hand side of (3.4), we obtain

$$\begin{aligned}\mathbb{E}_w\left[\pi^M\left(x_k^i|x_{k-1}, w_{k-1}\right)\right] &= \sum_{j=1}^M \left(\frac{1}{M}\sum_{j'=1}^M \delta_{w_{k-1}^j}(x_k^i|x_{k-1}, w_{k-1}^{j'})\right)\\ &= \frac{1}{M}\sum_{j=1}^M \delta_{w_{k-1}^j}\left(x_k^i|x_{k-1}, w_{k-1}^j\right),\end{aligned} \tag{3.6}$$

then replacing $\mathbb{E}_w[\pi^M(x_k^i|x_{k-1}, w_{k-1})]$ in (3.5) with (3.6), we have

$$\begin{aligned}p(x_k^i|Y_{1:k-1}) &= \int_{\mathbb{R}^d}\left[\frac{1}{M}\sum_{j=1}^M \delta_{w_{k-1}^j}\left(x_k^i|x_{k-1}, w_{k-1}^j\right)\right] p(x_{k-1}|Y_{1:k-1})dx_{k-1} + \mathcal{R}_{k|k-1}^i\\ &= \frac{1}{M}\sum_{j=1}^M p\left(x_{k-1}^{i,j}\middle|Y_{1:k-1}\right) + \mathcal{R}_{k|k-1}^i,\end{aligned} \tag{3.7}$$

where $p(x_{k-1}^{i,j}|Y_{1:k-1})$ is the value of $p\left(x_{k-1}|Y_{1:k-1}\right)$ at $x_{k-1}^{i,j}$. Neglecting the error term $\mathcal{R}_{k|k-1}^i$ in (3.7), we obtain the following iterative numerical scheme for constructing an approximation, denoted by $\varrho(x_k^i|Y_{1:k-1})$, of the prior PDF $p(x_k^i|Y_{1:k-1})$, i.e.,

$$\varrho(x_k^i|Y_{1:k-1}) = \frac{1}{M}\sum_{j=1}^M \varrho(x_{k-1}^{i,j}|Y_{1:k-1}). \tag{3.8}$$

In our previous work [2], the subsets $\mathcal{H}_k$, for $k = 0, 1, \ldots$, were defined by a full tensor product mesh, denoted by

$$\mathcal{M} := \mathcal{M}^{(1)} \times \mathcal{M}^{(2)} \times \cdots \mathcal{M}^{(d)}, \tag{3.9}$$

on a d-dimensional hyper-cube $[a_1, b_1] \times \cdots \times [a_d, b_d]$, where $\mathcal{M}^{(m)}, m = 1, \ldots, d$, is a uniform partition of the interval $[a_m, b_m]$ with $N^{(m)}$ grid points. It is simple to implement but has several significant disadvantages. First, at each time step, one needs to approximate the prior PDF $p(x_k|Y_{1:k-1})$ at a total of $N^{(1)} \times \cdots \times N^{(d)}$ grid points which grows exponentially as the dimension d increases. This is also known as "the curse of dimensionality". On the other hand, since the construction of $\mathcal{M}$ is not informed by the target PDF, the domain $[a_1, b_1] \times \cdots \times [a_d, b_d]$ needs to be defined sufficiently large, so as to capture the statistically significant region of the PDF. This may lead to a great waste of computation effort in the low probability region of $p(x_k|Y_{1:k-1})$.

To alleviate such disadvantages, we propose to develop a distribution-informed meshfree interpolation approach to efficiently approximate the prior PDF. The central idea of the generation of random points for the state variable is to build a set of points, denoted by $\mathcal{H}_k$, according to the state distribution. To begin with, we generate $\mathcal{H}_0 = \{\xi^i\}_{i=1}^N$ of N random samples from the initial PDF p_0 of the initial state:

$$\mathcal{H}_0 := \{x_0^i\}_{i=1}^N = \{\xi^i\}_{i=1}^N, \text{ with } x_0^i = \xi^i.$$

If the initial PDF p_0 is close to the true state distribution, it's obvious that our random state points are more concentrated near the target state. For $k = 1, 2, \cdots, K$, we propagate points $\{x_{k-1}^i\}_{i=1}^N$ to $\{x_k^i\}_{i=1}^N$ through the state equation (1.2):

$$x_k^i = f_{k-1}(x_{k-1}^i, \tilde{w}_{k-1}^i), \quad i = 1, 2, \cdots, N,$$

where $\{\tilde{w}_{k-1}^i\}_{i=1}^N$ are N random samples according to the PDF of w_{k-1}. Denote $\mathcal{H}_k := \{x_k^i\}_{i=1}^N$ and approximate the conditional PDF $p(x_k|Y_{1:k-1})$ on $\mathcal{H}_k$ with the scheme given by (3.8). In this way, the random points in $\mathcal{H}_k$ move according to the state model. As opposed to particle filter methods, which use the number of particles to represent empirical distributions and require a large number of particles to follow the state distribution, in the implicit filter method we provide an approximation of the value of the PDF at each state point. Therefore, much fewer points are needed to describe the state PDF and the random state points are not necessary to accurately follow the state distribution.

3.2. The update stage. By incorporating the new data Y_k, we update the prior PDF $p(x_k|Y_{1:k-1})$ at each grid point x_k^i, using the Bayesian formula, to obtain

$$\begin{aligned} p(x_k^i|Y_{1:k}) &= \frac{1}{C_k} p(Y_k|x_k^i) p(x_k^i|Y_{1:k-1}) \\ &= \frac{1}{C_k} p(Y_k|x_k^i) \varrho(x_k^i|Y_{1:k-1}) + \mathcal{R}_{k|k}^i, \end{aligned} \tag{3.10}$$

where $\varrho(x_k^i|Y_{1:k-1})$ is given in (3.8), C_k is the normalization factor, and $\mathcal{R}_{k|k}^i := \frac{1}{C_k} p(Y_k|x_k^i)\big(p(x_k^i|Y_{1:k-1}) - \varrho(x_k^i|Y_{1:k-1})\big)$ is the approximation error. By neglecting the error term $\mathcal{R}_{k|k}^i$ in (3.10), we obtain the following iterative numerical scheme for the update stage on $\mathcal{H}_k$, i.e.,

$$\varrho(x_k^i|Y_{1:k}) = \frac{1}{C_k} p(Y_k|x_k^i) \varrho(x_k^i|Y_{1:k-1}), \tag{3.11}$$

where $\varrho(x_k^i|Y_{1:k})$ is desired the approximation of the posterior PDF $p(x_k^i|Y_{1:k})$.

Next, we use interpolation methods to construct the approximation $\varrho(x_k|Y_{1:k})$ of $p(x_k|Y_{1:k})$ from values $\{\varrho(x_k^i|Y_{1:k})\}_{x_k^i \in \mathcal{H}_k}$ via

$$\varrho(x_k|Y_{1:k}) = \sum_{x_k^i \in \mathcal{H}_k} \varrho(x_k^i|Y_{1:k}) \phi^i(x_k), \tag{3.12}$$

where $\{\phi^i\}_{i=1}^{N_k}$ is the set of basis functions. Since the state points in $\mathcal{H}_k$ are generated randomly in the meshfree framework, standard polynomial interpolation [2] is unstable due to the uncontrollable Lebesgue constant. Instead, we propose to use the Shepard's method [12], which is an efficient meshfree interpolation technique, to construct the interpolant $\varrho(x_k|Y_{1:k})$. The basic idea of the Shepard's method is to use the weighted average of $\{\varrho(x_k^i|Y_{1:k})\}_{x_k^i \in \mathcal{H}_k}$ in the interpolating approximation. Specifically, for a given point $x_k \in \mathbb{R}^d$, we re-order the points in $\mathcal{H}_k$ by the distances to x_k to get a sequence $\{x_k^{(l)}\}_{l=1}^{N_k}$ such that

$$\|x_k - x_k^{(l_1)}\| \leq \|x_k - x_k^{(l_2)}\|, \text{ if } l_1 < l_2,$$

where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^d$. Then, for a pre-chosen integer L we use the first L values in $\{\varrho(x_k^{(l)}|Y_{1:k})\}_{l=1}^{N}$ to approximate $\varrho(x_k|Y_{1:k})$ as follows

$$\varrho(x_k|Y_{1:k}) = \sum_{l=1}^{L} \varrho(x_k^{(l)}|Y_{1:k}) \cdot h_l(x_k), \tag{3.13}$$

where the weight $h_l(x_k)$ is defined by

$$h_l(x_k) := \frac{\|x_k - x_k^{(l)}\|}{\sum_{l=1}^{L}\|x_k - x_k^{(l)}\|}.$$

Note that $\sum_{l=1}^{L} h_l(x_k) = 1$. From (3.13), we have

$$\begin{aligned}\varrho(x_k|Y_{1:k}) - p(x_k|Y_{1:k}) = & \sum_{l=1}^{L}\Big(\varrho(x_k^{(l)}|Y_{1:k}) - p(x_k^{(l)}|Y_{1:k})\Big) \cdot h_l(x_k) \\ & + \sum_{l=1}^{L} p(x_k^{(l)}|Y_{1:k}) \cdot h_l(x_k) - p(x_k|Y_{1:k}),\end{aligned}$$

where

$$\sum_{l=1}^{L} p(x_k^{(l)}|Y_{1:k}) \cdot h_l(x_k) - p(x_k|Y_{1:k}) = \sum_{l=1}^{L}\Big(p(x_k^{(l)}|Y_{1:k}) - p(x_k|Y_{1:k})\Big) \cdot h_l(x_k) \tag{3.14}$$

is the error of the Shepard's interpolation. We assume that $p(x_k|Y_{1:k})$ has bounded first order derivative. For each pair $p(x_k|Y_{1:k})$ and $p(x_k^{(l)}|Y_{1:k})$ the approximation error $|p(x_k^{(l)}|Y_{1:k}) - p(x_k|Y_{1:k})|$ is controlled by the distance $\|x_k - x_k^{(l)}\|$ and the derivative $p'(z|Y_{1:k})$, where z is a point between x_k and $x_k^{(l)}$. It is reasonable to assume that in high probability region of the derivative $p'(z|Y_{1:k})$ is large. It's worth pointing out that the random state points generated in this algorithm are concentrated in the high probability region. Thus, if x_k lies in the high probability region, the distance $\|x_k - x_k^{(l)}\|$ is small, which balances the error brought by the large derivative. On the other hand, if x_k lies in the low probability region, although the distance $\|x_k - x_k^{(l)}\|$ is relatively large, the approximation error (3.14) is still small due to the small value of the derivative $p'(z|Y_{1:k})$.

3.3. Resampling. Similar to the particle filter method, the above random state points generation suffers from the degeneracy problem for long term simulations, especially for high-dimensional problems. After several time steps, the probability density tends to concentrate on a few points which dramatically reduces the number of effective sample points in $\mathcal{H}_k$.

In this work, we propose an occasional resampling procedure to address these problems and rejuvenate the random points cloud. At the time step $k-1$, the resampling procedure takes place after we obtain $\varrho(x_{k-1}|Y_{1:k-1})$, in order to remove the degenerated points in $\mathcal{H}_{k-1}$ using the information provided by $\varrho(x_{k-1}|Y_{1:k-1})$. Specifically, the first step is to develop a degeneracy metric to determine the necessity of doing resampling. To this end, we define the following degenerated subset $\mathcal{S}_{k-1} \subset \mathcal{H}_{k-1}$,

$$\mathcal{S}_{k-1} = \left\{x^i_{k-1}|x^i_{k-1} \in \mathcal{H}_{k-1}, \varrho(x^i_{k-1}|Y_{1:k-1}) < \varepsilon\right\}, \tag{3.15}$$

where $\varepsilon > 0$ is a user-defined threshold. We also define

$$\mathcal{J}(\mathcal{S}_{k-1}) := \{i = 1, \ldots, N|x^i_{k-1} \in \mathcal{S}_{k-1}\}$$

to be the index set of $\mathcal{S}_{k-1}$. Then, the degeneracy of $\mathcal{H}_{k-1}$ can be measured by the ratio $\#(\mathcal{S}_{k-1})/\#(\mathcal{H}_{k-1}) \in [0, 1]$, where $\#(\cdot)$ denotes the number of points in a set. If the ratio is smaller than a threshold $\tau \in [0, 1]$, then we will skip the resampling step and propagate $\mathcal{H}_{k-1}$ to get $\mathcal{H}_k$; otherwise, the set $\mathcal{H}_{k-1}$ is considered degenerated, and the resampling procedure is needed.

In resampling, instead of propagating $\mathcal{H}_{k-1}$ to $\mathcal{H}_k$, we aim at constructing an intermediate point set, denoted by $\mathcal{H}_{k-\frac{1}{2}} := \{x^i_{k-\frac{1}{2}}\}_{i=1}^N$ and propagate $\mathcal{H}_{k-\frac{1}{2}}$ through the state model (1.2) to obtain $\mathcal{H}_k$. According to the definition of $\mathcal{S}_{k-1}$ in (3.15), we consider the state points in $\mathcal{H}_{k-1}\backslash\mathcal{S}_{k-1}$ are in the statistically significant region of $\varrho(x_{k-1}|Y_{1:k-1})$, so that we first put those points in $\mathcal{H}_{k-\frac{1}{2}}$, i.e.,

$$x^i_{k-\frac{1}{2}} = x^i_{k-1} \quad \text{for} \quad i \notin \mathcal{J}(\mathcal{S}_{k-1}).$$

For the state points in $\mathcal{S}_{k-1}$, we replace them by generating new samples from $\varrho(x_{k-1}|Y_{1:k-1})$ using the importance sampling [6], i.e.,

$$x^i_{k-\frac{1}{2}} \sim \varrho(x_{k-1}|Y_{1:k-1}) \quad \text{for} \quad i \in \mathcal{J}(\mathcal{S}_{k-1}).$$

As a result, the resampling procedure helps us remove the state points with low probabilities, and makes the state point set $\mathcal{H}_k$ concentrated in the high probability region of the posterior PDF $\varrho(x_{k-1}|Y_{1:k-1})$ at each time step.

3.4. Summary of the algorithm. Finally, we summarize the entire meshfree implicit filter algorithm introduced in §3.1-§3.3 in Algorithm 1 below.

4. Numerical experiments.

In this section, we present two numerical examples to examine the performance of our meshfree implicit filter method. In Example 1, we use a two dimensional nonlinear filtering problem to show the distributions of the random points $\mathcal{H}_k$. In Example 2, we solve a three dimensional bearing-only tracking problem, which is a six dimensional nonlinear filtering problem. For this higher dimensional problem, we compare the accuracy and efficiency of our meshfree implicit filter method with the extended Kalman filter and the particle filter.

Example 1. In this example, we consider the two dimensional noise perturbed tumoral growth model [21]

$$d\boldsymbol{X}_t = F(\boldsymbol{X}_t)dt + \boldsymbol{\sigma} \cdot dW_t, \tag{4.1}$$

where W_t is a two dimensional standard Brownian motion and $\boldsymbol{\sigma} = (0.01, 0.01)^T$. The state process $\boldsymbol{X}_t = (X^1_t, X^2_t)^T$ is a two dimensional vector, $F(\boldsymbol{X}_t) := (f_1(\boldsymbol{X}_t), f_2(\boldsymbol{X}_t))^T$ is defined as

$$f_1(\boldsymbol{X}_t) = \alpha_1 X^1_t \cdot \ln(\frac{X^2_t}{X^1_t})$$

Algorithm 1: *The meshfree implicit filter algorithm*

```
**Initialization**: set the number of samples $M$ for estimating $\mathbb{E}_w[\cdot]$, the number of state points $N$, the resampling thresholds $\varepsilon$ and $\tau$
while $k = 1, 2, \cdots$ , do
    Compute the ratio $\#(\mathcal{S}_{k-1})/\#(\mathcal{H}_{k-1})$
    if $\#(\mathcal{S}_{k-1})/\#(\mathcal{H}_{k-1}) < \tau$ then
        Propagate $\mathcal{H}_{k-1}$ through the state model (1.2) to obtain $\mathcal{H}_k$
    else
        Resample and construct the intermediate state set $\mathcal{H}_{k-\frac{1}{2}}$
        Propagate $\mathcal{H}_{k-\frac{1}{2}}$ through the state model (1.2) to obtain $\mathcal{H}_k$
    end if
    Prediction: solve $\varrho(x_k|Y_{1:k-1})$ using (3.8), at each point in $\mathcal{H}_k$
    Update: solve $\varrho(x_k|Y_{1:k})$ using (3.11) and (3.13)
end while
```

and

$$f_2(\boldsymbol{X}_t) = \alpha_2 X_t^1 - \alpha_3 X_t^2 \cdot (X_t^1)^{\frac{2}{3}}.$$

Here, f_1 models the Gompertzian growth rate of the tumor and f_2 gives the degree of vascularization of the tumor which is also called " angiogenic capacity".

To approximate the state variables, we discretize the dynamic system (4.1) in time and obtain a discrete state model

$$\boldsymbol{X}_k = F(\boldsymbol{X}_{k-1}) \cdot \Delta + \boldsymbol{\sigma} \cdot \boldsymbol{\omega}_{k-1}. \tag{4.2}$$

Here, $\boldsymbol{\omega}_k$ is a two dimensional zero mean Gaussian white noise process with covariance $Q = I\Delta$, where I is the 2×2 identity matrix and Δ is the time partition stepsize.

The measurement of the state model is given by

$$Y_k = \left(X_k^1, X_k^2\right)^T + \boldsymbol{R} \cdot \boldsymbol{v}_k,$$

where $\boldsymbol{v}_k$ is a two dimensional zero mean Gaussian white noise process with covariance $\Lambda = I\Delta$, I is a 2×2 identity matrix and $\boldsymbol{R} = (0.1, 0.1)^T$.

In the numerical experiment, we use uniform time partition with stepsize $\Delta = 0.2$ and simulate the state process for $K = 40$ with initial state $\boldsymbol{X}_0 = (0.8, 0.3)^T$ and parameters $\alpha_1 = 1$, $\alpha_2 = 0.2$, $\alpha_3 = 0.2$. At time step $k = 0$, we initialize the prior PDF p_0 by $N(\tilde{\boldsymbol{X}}_0, \Sigma)$, where $\tilde{\boldsymbol{X}}_0 = (0.78, 0.32)^T$ and

$$\Sigma = \begin{pmatrix} 0.05^2 & 0 \\ 0 & 0.1^2 \end{pmatrix}. \tag{4.3}$$

In Figure 4.1, we plot 1500 random samples generated from the initial PDF p_0, which are our initial random points $\mathcal{H}_0$. Figure 4.2 illustrates the behavior of random state points $\mathcal{H}_k$ at time steps $k = 1, 2, 3, 10, 20, 40$, respectively. In Figure 4.2, the blue dots in each figure plot the random state points obtained by using the dynamic

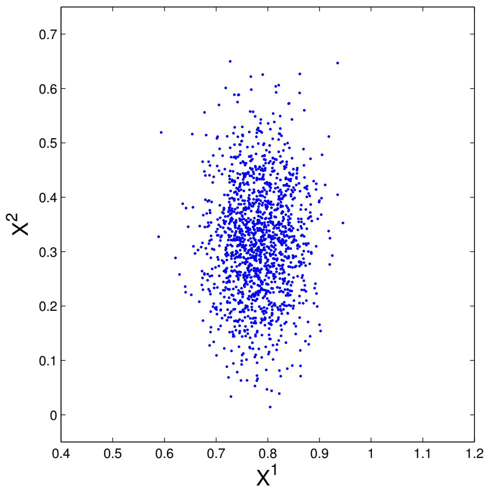


Fig. 4.1: Example 1 : Initial random state space points $\mathcal{H}_0$

state points generation method introduced in Section 3 and the red cross in each figure gives the true state $\boldsymbol{X}_k$ at the corresponding time step. From the figures we can see that all the points are moving according to the state model and are concentrated around the true state.

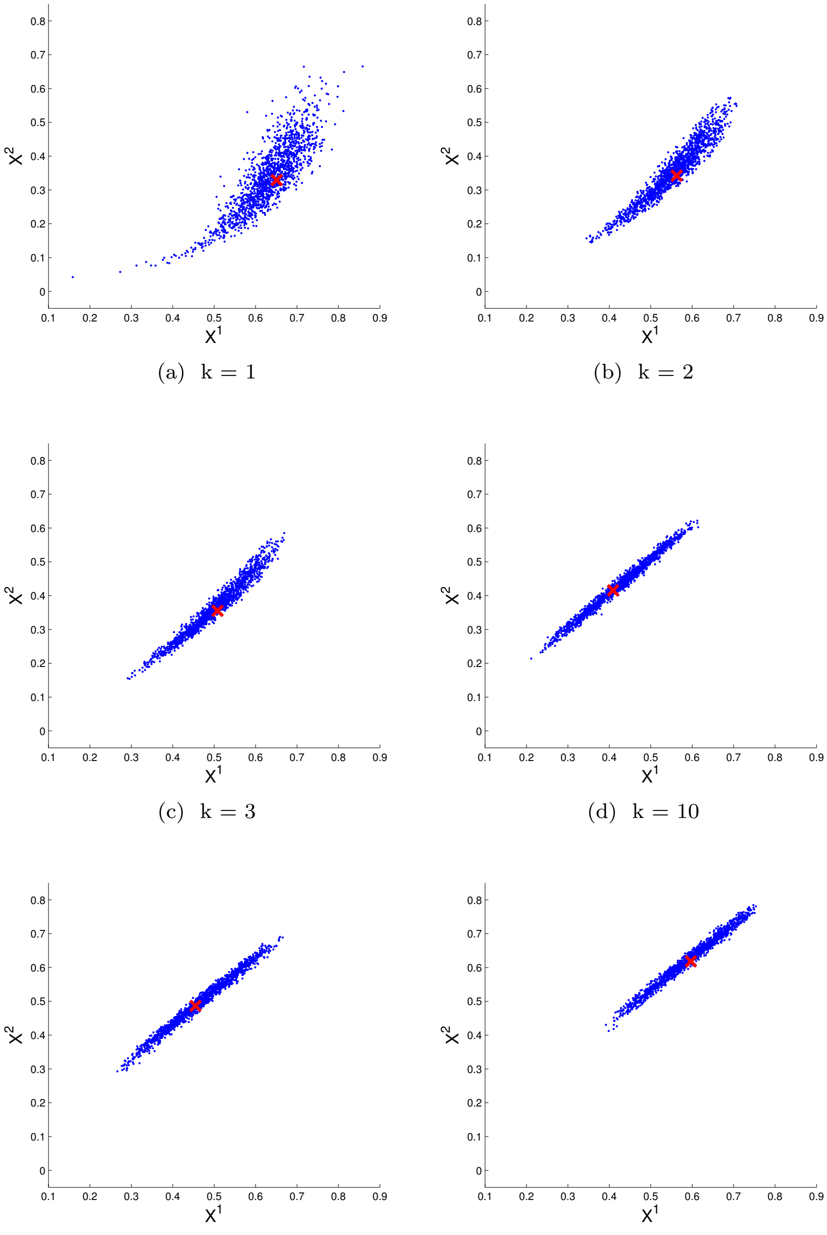
To present the accuracy of the algorithm, we show the simulation of the tumoral growth states in Figure 4.3. The black curves are the true X^1 and X^2 coordinate values of the tumoral growth states, respectively. The blue curves show the simulated states obtained by using the meshfree implicit filter method.

Example 2. In this example, we study a six dimensional target tracking problem. In Figure 4.4, the target, denoted by the red line, moves in the three dimensional space and two platforms on the ground, denoted by pentagons, take angular observations of the moving target.

The state process $\boldsymbol{X}_k = (X_k^1, X_k^2, X_k^3, X_k^4, X_k^5, X_k^6)^T$ is described by the following dynamic model

$$\boldsymbol{X}_k = f(\boldsymbol{X}_{k-1}) + \boldsymbol{\sigma} \cdot \boldsymbol{\omega}_{k-1}, \tag{4.4}$$

where (X^1, X^2, X^3) describes the position of the moving target which is controlled by parameters (X^4, X^5, X^6). The system noise $\boldsymbol{\omega_k} = (\omega_k^1, \omega_k^2, \omega_k^3, \omega_k^4, \omega_k^5, \omega_k^6)^T$ is a zero mean Gaussian white noise process with covariance $Q \doteq I\Delta$, I is the 6×6 identity matrix and Δ is a given time period, $\boldsymbol{\sigma} = (0.1, 0.1, 0.1, 0.01, 0.01, 0.01)^T$ is a constant



(a) k = 1

(b) k = 2

(c) k = 3

(d) k = 10

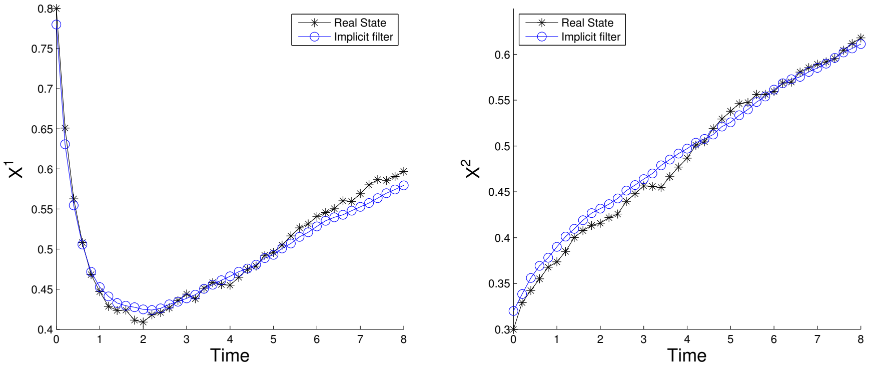
(e) k = 20

(f) k = 40

Fig. 4.2: Example 1: State space points $\mathcal{H}_k$ at time steps $k = 1, 2, 3, 10, 20, 40$.

vector and f is given by

$$f(\boldsymbol{X}_k) = \begin{pmatrix} X^1_{k-1} + X^4_{k-1}\Delta \\ X^2_{k-1} + \sin(\alpha X^5_{k-1})\Delta \\ X^3_{k-1} + (X^6_{k-1})^2\Delta \\ X^4_{k-1} + v_1\Delta \\ X^5_{k-1} + v_2\Delta \\ X^6_{k-1} + v_3\Delta \end{pmatrix}.$$



(a) Simulation: X^1

(b) Simulation: X^2

Fig. 4.3: Example 1: Simulation of the tumoral growth states

The measurements $\boldsymbol{Y}_k$ of the state process from the two locations are given by

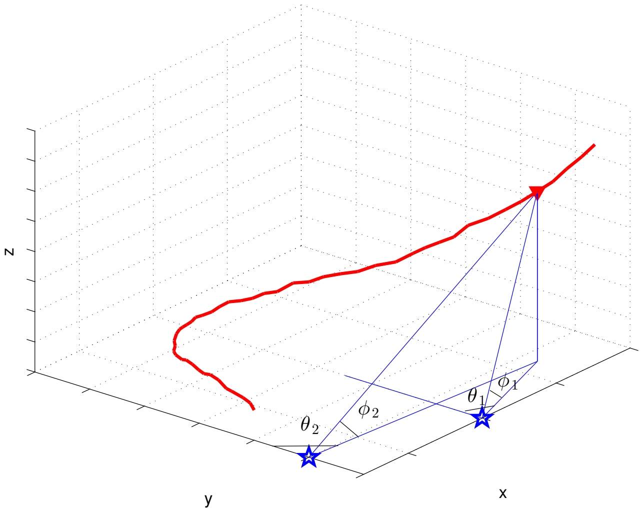


Fig. 4.4: Example 2: Bearing-only Tracking in 3-D

$$\boldsymbol{Y}_k = \begin{pmatrix} \arctan\left(\frac{X_k^3}{\sqrt{(X_k^1-a_1)^2+(X_k^2-b_1)^2}}\right) \\ \arctan\left(\frac{X_k^3}{\sqrt{(X_k^1-a_2)^2+(X_k^2-b_2)^2}}\right) \\ \arctan\left(\frac{X_k^1-a_1}{X_k^2-b_1}\right) \\ \arctan\left(\frac{X_k^1-a_2}{X_k^2-b_2}\right) \end{pmatrix} + \boldsymbol{R}\boldsymbol{v}_k,$$

where $\boldsymbol{v}_k$ is a 4 dimensional zero mean Gaussian white noise process with covariance $\Lambda = I\Delta$, I is a 4×4 identity matrix, $\boldsymbol{R} = (0.6, 0.6, 0.6, 0.6)^T$, (a_1, b_1) and (a_2, b_2) are locations of two observers. We choose $\Delta = 0.3$, $\alpha = 3$, $v_1 = v_2 = v_3 = 0.05$. Also, we assume that platforms are located at $(a_1, b_1) = (16, 6)$, $(a_2, b_2) = (8, 15)$ and the initial sate is given by a Gaussian $N(\boldsymbol{X}_0, \Sigma)$ where $\boldsymbol{X}_0 = (2, 2, 1, 0.4, 0.4, 0)^T$ and

$$\Sigma = \begin{pmatrix} 1^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2^2 \end{pmatrix}.$$

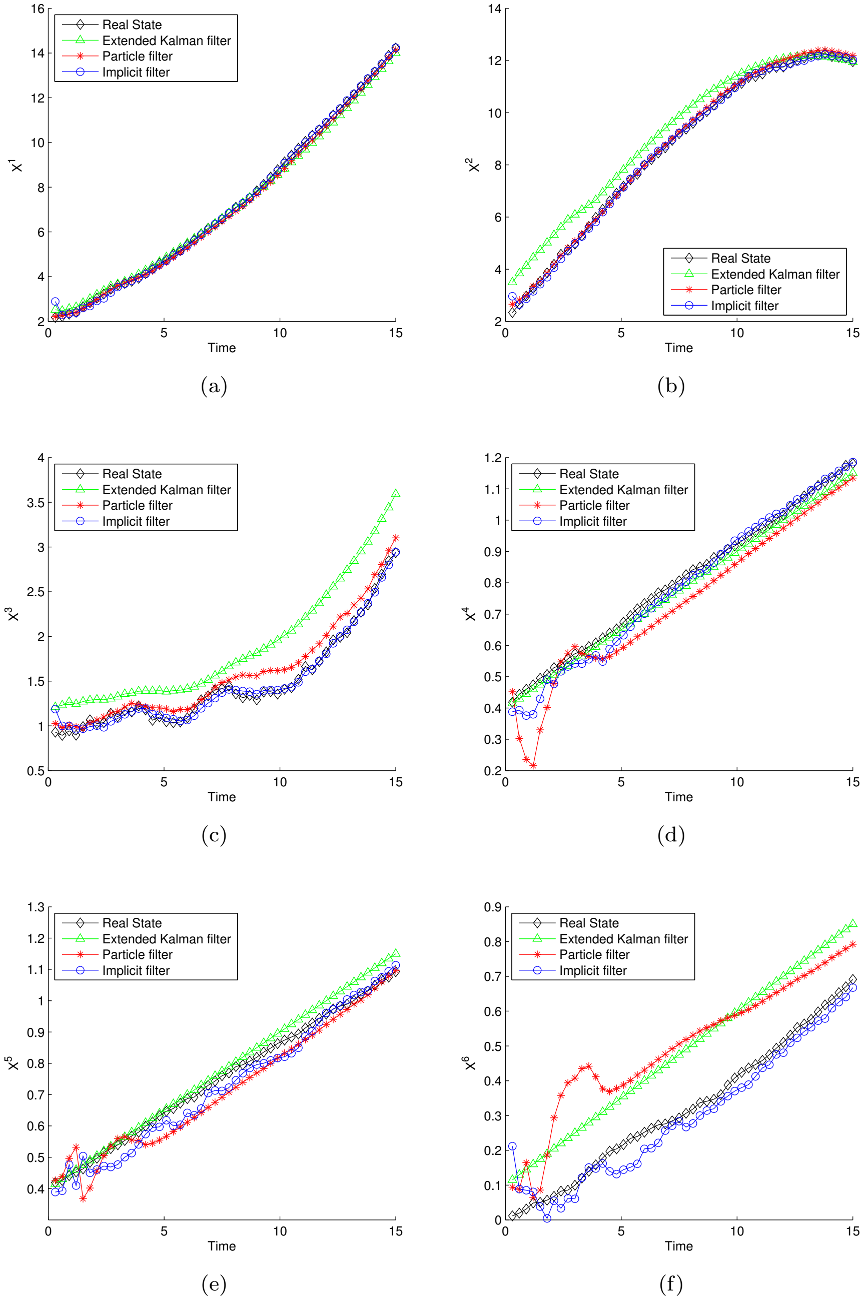
The target will be observed over the time period $0 \leq t \leq 15$. In the numerical experiments, we compare the performance of our meshfree implicit filter with the extended Kalman filter and the particle filter. In particular, we compare the estimated mean values of the states process along each dimension in Figure 4.5. In the particle filter method, we choose $15,000$ particles. In the meshfree implicit filter method, we choose the number of state points to be $N = 4,000$ and the number of random samples in the implicit filter Monte Carlo simulation to be $M = 6$. The black curves in Figure 4.5 show the real states process along each direction, the green curves give the estimated means obtained by the extended Kalman filter method, the red curves give the estimated means obtained by the particle filter method, and the blue curves give the estimated means obtained by the meshfree implicit filter. We also plot the L^2 error err_k corresponding to all three methods in figure 4.6. As we can see from figure 4.5 and 4.6, the implicit filter and the particle filter are much more accurate than the extended Kalman filter and the implicit filter is the most accurate approximation in this experiment.

To further compare the efficiency between the meshfree implicit filter and the particle filter, we repeat the above experiment over 50 realizations and show the average CPU time and the corresponding global root mean square error err_G defined by

$$err_G^2 = \frac{1}{50}\frac{1}{K}\sum_{j=1}^{50}\sum_{k=1}^{K} err_k(j)^2$$

where $err_k(j)$ is the L^2 error of the j-realization at time step k. In table 4.1, we can see that with $15,000$ particles, the CPU time of the particle filter method is comparable to that of the implicit filter with $4,000$ random state points, but the global RMSE of the particle filter is more than doubled the RMSE of the implicit filter. With $25,000$ particles, the particle filter method achieves an accuracy comparable to the implicit filter, but at a significantly higher cost.

5. Conclusions. In this work, we proposed an efficient meshfree implicit filter algorithm by evaluating the conditional PDF on meshfree points in the state space. These meshfree points are chosen adaptively according to the system state evolution. We also apply Shepard's method as the meshfree interpolation method to compute interplants with random state points. In order to address the degeneracy of the random points, we use importance sampling method to construct a resample step. Numerical examples demonstrate the effectiveness and efficiency of our algorithm. In



(a) (b) (c) (d) (e) (f)

Fig. 4.5: Example 2 : Comparison of estimated states. (a) Shows the comparison on X^1 direction. (b) Shows the comparison on X^2 direction. (c) Shows the comparison on X^3 direction. (d) Shows the comparison on X^4 direction. (e) Shows the comparison on X^5 direction. (f) Shows the comparison on X^6 direction.

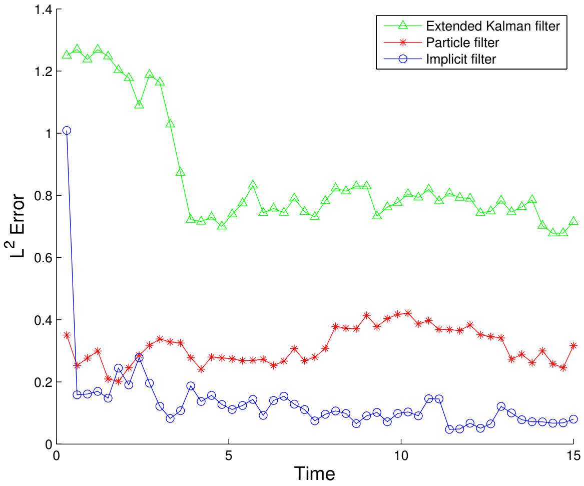


Fig. 4.6: Example 2 : Comparison of L^2 error.

Table 4.1: Example 2: Efficiency comparison

Methods	CPU time (seconds)	err_G
Implicit filter ($4,000$ state points)	83.14	0.0924
Particle filter ($15,000$ particles)	82.89	0.2545
Particle filter ($20,000$ particles)	142.61	0.1687
Particle filter ($25,000$ particles)	209.27	0.1057

the future, we plan to perform a rigorous numerical analysis for the meshfree implicit filter algorithm.

REFERENCES

[1] Syed Baker, Hart Poskar, Falk Schreilber, and Bjorn Junker. An improved constraint filtering technique for inferring hidden states and parameters of a biological model. *Bioinformatics*, 29:1052–1059, 2013.

[2] F. Bao, Y. Cao, and X. Han. An implicit algorithm of solving nonlinear filtering problems. *Commun. Comput. Phys.*, 16(2):382–402, 2014.

[3] Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking and data association*, volume 179 of *Mathematics in Science and Engineering*. Academic Press Inc., San Diego, CA, 1988.

[4] Alain Bensoussan, Jussi Keppo, and Suresh P. Sethi. Optimal consumption and portfolio decisions with partially observed real prices. *Math. Finance*, 19(2):215–236, 2009.

[5] Miodrag Bolić, Petar M. Djurić, and Sangjin Hong. Resampling algorithms and architectures for distributed particle filters. *IEEE Trans. Signal Process.*, 53(7):2442–2450, 2005.

[6] Amarjit Budhiraja, Lingji Chen, and Chihoon Lee. A survey of numerical methods for nonlinear filtering problems. *Phys. D*, 230(1-2):27–36, 2007.

[7] D. Crisan and O. Obanubi. Particle filters with random resampling times. *Stochastic Process. Appl.*, 122(4):1332–1368, 2012.

[8] Dan Crisan and Jie Xiong. A central limit type theorem for a class of particle filters. *Commun. Stoch. Anal.*, 1(1):103–122, 2007.

[9] Crisan D. and Doucet A. A survey of convergence results on particle filtering methods for practitiners. *IEEE Trans. Sig. Proc.*

[10] Jindřich Duník, Miroslav Šimandl, and Ondřej Straka. Unscented Kalman filter: aspects and adaptive setting of scaling parameter. *IEEE Trans. Automat. Control*, 57(9):2411–2416, 2012.

[11] Robert J. Elliott and Tak Kuen Siu. Option pricing and filtering with hidden Markov-modulated pure-jump processes. *Appl. Math. Finance*, 20(1):1–25, 2013.

[12] Gregory F. Fasshauer. *Meshfree Approximation Methods with MATLAB*. Interdisciplinary Mathematical Sciences (Book 6). World Scientific Publishing Company, 2007.

[13] N.J Gordon, D.J Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE PROCEEDING-F*, 140(2):107–113, 1993.

[14] M. Hairer, A. Stuart, and J. Voss. Signal processing problems on function space: Bayesian formulation, stochastic PDEs and effective MCMC methods. *The Oxford handbook of nonlinear filtering*, pages 833–873, 2011.

[15] G. Huang and P. Mehta. Joint probabilistic data association-feedback particle filter with applications to multiple target tracking. *Procs. of American Control Conference*, 2012.

[16] Adrew H. Jazwinski. *Stochastic Processing and Filtering Theory*, volume 64. Academic Press, New York, 1973.

[17] Simon J. Julier and Joseph J. LaViola, Jr. On Kalman filtering with nonlinear equality constraints. *IEEE Trans. Signal Process.*, 55(6, part 2):2774–2784, 2007.

[18] D. Kim, T. Song, and D. Musicki. Highest probability data association for multi-target particle filtering with nonlinear measurements. *IEICE TRANSACTIONS on Communications*, E96-B (1):281–290, 2013.

[19] Gennady Yu. Kulikov and Maria V. Kulikova. Accurate numerical implementation of the continuous-discrete extended Kalman filter. *IEEE Trans. Automat. Control*, 59(1):273–279, 2014.

[20] Max A. Little and Nick S. Jones. Signal processing for molecular and cellular biological physics: an emerging field. *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 371(1984):20110546, 18, 2013.

[21] Benzekry Sebastien. Mathematical analysis of a two-dimensional population model of metastatic growth including angiogenesis. *J. Evol. Equ.*, 11:187–213, 2011.

[22] S. Singh, S. Digumarthy, A. Back, J. Shepard, and Kalra M. Radiation dose reduction for chest ct with non-linear adaptive filters. *Acta Radiologica*, 55(2):169–174, 2013.

[23] W. Stannat. Stability of the optimal filter for nonergodic signals—a variational approach. *The Oxford handbook of nonlinear filtering*, pages 374–399, 2011.